

REMARKS

Upon entry of this amendment, claims 32, 42, 48-50, 60, 62, 64-66, 68 and 69 are all the claims pending in the application. Claims 19-22, 30, 33, 43, 51, 55-59, 61, 63 and 67 have been canceled by this amendment, and claims 68 and 69 have been added as new claims. No new matter has been added.

Applicants would like to thank Examiner Zheng Wei for the courtesies extended to Applicants' representative during the telephone interview conducted on February 25, 2009. During the interview, proposed claim changes were discussed for distinguishing the claims over the references applied in the Office Action. In this regard, Applicants note that the claim changes presented herein generally correspond to the claim changes discussed during the interview. Applicants respectfully submit that the claims as amended herein are patentable over the cited prior art references for the reasons discussed below.

I. Claim Rejections under 35 U.S.C. § 102

Claims 30, 32, 33, 42 and 43 have been rejected under 35 U.S.C. § 102(b) as being anticipated by Stallman (Using and Porting the GNU Compiler Collection for GCC 3.1).

Regarding claim 32, Applicants note that this claim has been amended to recite that the directive acquisition unit detects a directive for guaranteeing that data indicated by a pointer variable shown by the name of a specific variable is aligned by a specific value in the source program, wherein the optimization unit performs the optimization assuming that the data indicated by the pointer variable that is an object of the directive detected by the directive

acquisition unit is allocated in the memory region by the guaranteed value of alignment.

Applicants respectfully submit that Stallman does not disclose or suggest at least the above-noted features recited in amended claim 32.

With respect to Stallman, Applicants note that this reference discloses a conventional compiler apparatus in which a directive is obtained which instructs the compiler apparatus how to compile the program. For example, regarding alignment, Stallman explains that a directive is obtained which instructs the compiler of the specific alignment that is to be used (see pages 181-182 of Stallman).

In contrast to the above-noted conventional disclosure of Stallman in which a directive for alignment instructs the compiler of the specific alignment that is to be used, according to claim 32 as amended herein, the directive guarantees that data indicated by a pointer variable shown by the name of a specific variable is aligned by a specific value in the source program, wherein the optimization unit performs the optimization assuming that the data indicated by the pointer variable that is an object of the directive detected by the directive acquisition unit is allocated in the memory region by the guaranteed value of alignment.

By providing the above-noted “guaranteed value of alignment”, Applicants note that the compiler apparatus according to the claimed invention is able to reduce the code size and increase execution speed.

In view of the foregoing, Applicants respectfully submit that Stallman does not disclose, suggest or otherwise render obvious at least the above-noted features recited in amended claim 32. Accordingly, Applicants submit that claim 32 is patentable over Stallman, an indication of

which is kindly requested. Claim 42, as well as new claims 68 and 69, depend from claim 32 and are therefore considered patentable at least by virtue of their dependency.

Regarding claims 30, 33 and 43, as noted above, these claims have been canceled by this amendment.

II. Claim Rejections under 35 U.S.C. § 103(a)

Claims 19-22, 48-51 and 55-67 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Stallman (Using and Porting the GNU Compiler Collection for GCC 3.1) in view of PGI (PGI Workstation User's Guide-9 Optimization Directive and Pragmas), and further in view of Geva (U.S. 6,539,541).

A. Claim 48

Regarding claim 48, Applicants note that this claim has been amended so as to recite that the directive acquisition unit detects a directive for guaranteeing that the number of iterations of specific loop processing in the source program is one value of a set of values, wherein the optimization unit restrains generation of an escape code that is needed when the guaranteed value of the set of values is a value equal to 1 or more. Applicants respectfully submit that the cited prior art references do not teach or suggest at least this feature of claim 48.

With respect to Stallman, as noted above, this reference discloses a conventional compiler apparatus in which a directive is obtained which instructs the compiler apparatus how to compile the program.

Thus, while Stallman discloses the ability to instruct the compiler how to compile a

program, Applicants respectfully submit that Stallman does not disclose or suggest the above-noted features recited in claim 48 which indicate that the directive acquisition unit detects a directive for guaranteeing that the number of iterations of specific loop processing in the source program is one value of a set of values, wherein the optimization unit restrains generation of an escape code that is needed when the guaranteed value of the set of values is a value equal to 1 or more.

With respect to PGI, Applicants note that this reference discloses the use of a directive which instructs a compiler whether or not to perform loop unrolling (see section 9.4, Table 9-2).

Thus, while PGI discloses the ability to instruct the compiler whether or not to perform loop unrolling, Applicants respectfully submit that PGI does not disclose or suggest the above-noted features recited in claim 48 which indicate that the directive acquisition unit detects a directive for guaranteeing that the number of iterations of specific loop processing in the source program is one value of a set of values, wherein the optimization unit restrains generation of an escape code that is needed when the guaranteed value of the set of values is a value equal to 1 or more.

With respect to Geva, Applicants note that this reference discloses a compiler which reads information indicating the number of loop iterations, and then performs loop processing for the indicated number of loop iterations (see col. 9, lines 9-50).

Thus, while Geva discloses the ability to read information indicating the number of loop iterations, and to merely perform loop processing for the indicated number, Applicants respectfully submit that Geva does not disclose or suggest the above-noted features recited in

claim 48 which indicate that the directive acquisition unit detects a directive for guaranteeing that the number of iterations of specific loop processing in the source program is one value of a set of values, wherein the optimization unit restrains generation of an escape code that is needed when the guaranteed value of the set of values is a value equal to 1 or more.

In view of the foregoing, Applicants respectfully submit that the cited prior art references do not teach, suggest or otherwise render obvious at least the above-noted features recited in amended claim 48. Accordingly, Applicants submit that claim 48 is patentable over the cited prior art, an indication of which is kindly requested.

B. Claim 49

Regarding claim 49, Applicants note that this claim has been amended so as to recite that the directive acquisition unit detects a directive for guaranteeing that the number of iterations of specific loop processing in the source program is one value of a set of values, wherein the optimization unit performs the optimization by loop unrolling when the guaranteed value of the set of values is a value equal to or more than the number of development by the loop unrolling.

Applicants respectfully submit that the cited prior art references do not teach or suggest at least this feature of claim 49.

With respect to Stallman, as noted above, this reference discloses a conventional compiler apparatus in which a directive is obtained which instructs the compiler apparatus how to compile the program.

Thus, while Stallman discloses the ability to instruct the compiler how to compile a program, Applicants respectfully submit that Stallman does not disclose or suggest the above-

noted features recited in claim 49 which indicate that the directive acquisition unit detects a directive for guaranteeing that the number of iterations of specific loop processing in the source program is one value of a set of values, wherein the optimization unit performs the optimization by loop unrolling when the guaranteed value of the set of values is a value equal to or more than the number of development by the loop unrolling.

With respect to PGI, Applicants note that this reference discloses the use of a directive which instructs a compiler whether or not to perform loop unrolling (see section 9.4, Table 9-2).

Thus, while PGI discloses the ability to instruct the compiler whether or not to perform loop unrolling, Applicants respectfully submit that PGI does not disclose or suggest the above-noted features recited in claim 49 which indicate that the directive acquisition unit detects a directive for guaranteeing that the number of iterations of specific loop processing in the source program is one value of a set of values, wherein the optimization unit performs the optimization by loop unrolling when the guaranteed value of the set of values is a value equal to or more than the number of development by the loop unrolling.

With respect to Geva, Applicants note that this reference discloses a compiler which reads information indicating the number of loop iterations, and then performs loop processing for the indicated number of loop iterations (see col. 9, lines 9-50).

Thus, while Geva discloses the ability to read information indicating the number of loop iterations, and to merely perform loop processing for the indicated number, Applicants respectfully submit that Geva does not disclose or suggest the above-noted features recited in claim 49 which indicate that the directive acquisition unit detects a directive for guaranteeing that

the number of iterations of specific loop processing in the source program is one value of a set of values, wherein the optimization unit performs the optimization by loop unrolling when the guaranteed value of the set of values is a value equal to or more than the number of development by the loop unrolling.

In view of the foregoing, Applicants respectfully submit that the cited prior art references do not teach, suggest or otherwise render obvious at least the above-noted features recited in amended claim 49. Accordingly, Applicants submit that claim 49 is patentable over the cited prior art, an indication of which is kindly requested.

C. Claim 50

Regarding claim 50, Applicants note that this claim has been amended so as to recite that the directive acquisition unit detects a directive for guaranteeing that the number of iterations of specific loop processing in the source program is one of a set of values, wherein the optimization unit performs the optimization by loop unrolling assuming that the loop processing that is an object of designation detected by the directive acquisition unit is iterated only the even number of times depending on whether the guaranteed set of values is a set of only even values or a set of only odd values.

Applicants respectfully submit that the cited prior art references do not teach or suggest at least this feature of claim 50.

With respect to Stallman, as noted above, this reference discloses a conventional compiler apparatus in which a directive is obtained which instructs the compiler apparatus how to compile the program.

Thus, while Stallman discloses the ability to instruct the compiler how to compile a program, Applicants respectfully submit that Stallman does not disclose or suggest the above-noted features recited in claim 50 which indicate that the directive acquisition unit detects a directive for guaranteeing that the number of iterations of specific loop processing in the source program is one of a set of values, wherein the optimization unit performs the optimization by loop unrolling assuming that the loop processing that is an object of designation detected by the directive acquisition unit is iterated only the even number of times depending on whether the guaranteed set of values is a set of only even values or a set of only odd values.

With respect to PGI, Applicants note that this reference discloses the use of a directive which instructs a compiler whether or not to perform loop unrolling (see section 9.4, Table 9-2).

Thus, while PGI discloses the ability to instruct the compiler whether or not to perform loop unrolling, Applicants respectfully submit that PGI does not disclose or suggest the above-noted features recited in claim 50 which indicate that the directive acquisition unit detects a directive for guaranteeing that the number of iterations of specific loop processing in the source program is one of a set of values, wherein the optimization unit performs the optimization by loop unrolling assuming that the loop processing that is an object of designation detected by the directive acquisition unit is iterated only the even number of times depending on whether the guaranteed set of values is a set of only even values or a set of only odd values.

With respect to Geva, Applicants note that this reference discloses a compiler which reads information indicating the number of loop iterations, and then performs loop processing for the indicated number of loop iterations (see col. 9, lines 9-50).

Thus, while Geva discloses the ability to read information indicating the number of loop iterations, and to merely perform loop processing for the indicated number, Applicants respectfully submit that Geva does not disclose or suggest the above-noted features recited in claim 50 which indicate that the directive acquisition unit detects a directive for guaranteeing that the number of iterations of specific loop processing in the source program is one of a set of values, wherein the optimization unit performs the optimization by loop unrolling assuming that the loop processing that is an object of designation detected by the directive acquisition unit is iterated only the even number of times depending on whether the guaranteed set of values is a set of only even values or a set of only odd values.

In view of the foregoing, Applicants respectfully submit that the cited prior art references do not teach, suggest or otherwise render obvious at least the above-noted features recited in amended claim 50. Accordingly, Applicants submit that claim 50 is patentable over the cited prior art, an indication of which is kindly requested.

D. Claims 60, 62 and 64-66

Regarding claims 60, 62 and 64-66, Applicants note that these claims recite similar features as set forth in claims 32, 42 and 48-50, respectively. Accordingly, for at least similar reasons as described above, Applicants respectfully submit that the cited prior art references do not disclose, suggest or otherwise render obvious all of the features recited in claims 60, 62 and 64-66. Accordingly, Applicants submit that claims 60, 62 and 64-66 are patentable over the cited prior art, an indication of which is kindly requested.

E. Claims 19-22, 51, 55-59, 61, 63 and 67

Regarding claims 19-22, 51, 55-59, 61, 63 and 67, as noted above, these claims have been canceled by this amendment.

III. Conclusion

In view of the above, reconsideration and allowance of this application are now believed to be in order, and such actions are hereby solicited. If any points remain in issue which the Examiner feels may best be resolved through a personal or telephone interview, the Examiner is kindly requested to contact the undersigned at the telephone number listed below.

Respectfully submitted,

Hajime OGAWA et al.

/Kenneth W. Fields/

By: 2009.03.02 15:09:17 -05'00'

Kenneth W. Fields
Registration No. 52,430
Attorney for Applicants

KWF/krq
Washington, D.C. 20005-1503
Telephone (202) 721-8200
Facsimile (202) 721-8250
March 2, 2009